



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

## Utility Patent Application

### **ADAPTIVE REGION EDITING TOOL**

Inventor(s):

**Krzysztof Antoni Zaklika**

**Mikhail Ivanovich Trifonov**

**Leonid Borisovich Lalyko**

**Viktor Anatolyevich Nesterov**

**Pavel Vladimirovich Gruzdev**

ATTORNEY'S DOCKET NO. JS1-005US

## **ADAPTIVE REGION EDITING TOOL**

### **Technical Field**

The invention relates generally to image editing, and more particularly to an adaptive region editing tool.

### **Description**

Digital image editing has made many sophisticated graphics and multimedia editing capabilities accessible to a large number of professionals and consumers. Users can edit digital images, such as photographs, video frames, computer graphics, etc., using reasonably-priced image editing software. Such software can execute on a variety of computers, including graphics workstations, desktop computers, and laptop computers. However, depending on the user's abilities, some of these sophisticated editing capabilities may still present tasks that are too complex or too time-consuming for many users to perform with acceptable results.

One common image editing operation involves the selective editing of a region in the image. For example, given a still digital image (e.g., a digital photograph or a video frame) of a human subject, a user may wish to remove (e.g., erase) the background in the image (e.g., an office environment) and replace it with a different background (e.g., a beach scene). In one approach, region erasure of the background can be accomplished manually, pixel-by-pixel, by changing the color and/or opacity of each pixel of the background. However, not only can this approach be tedious and very time-consuming, manually determining the appropriate color and translucence of each pixel on the boundary between the

1 subject and the background can be quite complicated. First, it can be difficult to  
2 correctly identify which pixels are actually in the human subject and which pixels  
3 are in the background, particularly when the subject has a complex shape or very  
4 fine attributes. Furthermore, some pixels include a blend of both subject and  
5 background and, therefore, may require some manual “unblending” of colors from  
6 the subject and the background to produce a satisfactory result. However, manual  
7 unblending can be extremely difficult and often unworkable for many users.  
8 Accordingly, the effort required to employ a manual approach may be  
9 unacceptable for the quality of the result achieved.

10 Some more automated approaches for editing a region can be used. For  
11 example, to erase a background around a subject in an image, a user may designate  
12 a clipping path that generally distinguishes the subject from the background. The  
13 region designated outside the clipping path is erased, and the boundary of the  
14 clipped region can be blurred or blended to substantially minimize the influence of  
15 the remaining background on the clipped subject. Although this clipping approach  
16 is less manual than a pixel-by-pixel approach, the approach typically relies on a  
17 user specifying the clipping path closely around the boundary of the subject in  
18 order to achieved acceptable results. The more background that remains in the  
19 clipped subject, the more influence the remaining background has on the resulting  
20 edited image – an undesirable effect. Furthermore, such blending or blurring tends  
21 to decrease the sharpness of the subject boundary, even though sharpness is  
22 frequently a primary desirable characteristic.

23 Other approaches employ automated visual blending techniques that do not  
24 require clipping. However, such existing approaches define an individual erasure  
25 color based on analysis of a single region of a tool impression. These approaches

1 fail to take sufficient advantage of the instructive effort of a user, who is capable  
2 of communicating useful information about different regions within a tool  
3 impression simply by his or her placement of the tool within the image.

4 Implementations described and claimed herein sample properties of pixels  
5 in a digital image within different subdivisions of an editing tool impression to  
6 produce different property distributions. In one implementation, the subdivisions  
7 are differently located within the tool impression. The property distributions from  
8 each region are classified to identify different edit classes within the property  
9 space, which are then used to apply an edit effect to the digital image within the  
10 tool impression. The edit classes may be represented by an edit profile in two or  
11 more dimensions (e.g., applying to one or more pixel properties).

12 In some implementations, articles of manufacture are provided as computer  
13 program products. One implementation of a computer program product provides a  
14 computer program storage medium readable by a computer system and encoding a  
15 computer program. Another implementation of a computer program product may  
16 be provided in a computer data signal embodied in a carrier wave by a computing  
17 system and encoding the computer program.

18 The computer program product encodes a computer program for executing  
19 a computer process on a computer system. Pixels in a first region within a tool  
20 impression in a digital image are sampled to determine a first distribution of a  
21 pixel property of the pixels in the first region. Pixels in a second region within the  
22 tool impression are sampled to determine a second distribution of the pixel  
23 property of the pixels in the second region. At least one pixel within the tool  
24 impression is edited based on the first and second distributions.  
25

1 In another implementation, a method is provided. Pixels in a first region  
2 within a tool impression in a digital image are sampled to determine a first  
3 distribution of a pixel property of the pixels in the first region. Pixels in a second  
4 region within the tool impression are sampled to determine a second distribution  
5 of the pixel property of the pixels in the second region. At least one pixel within  
6 the tool impression is edited based on the first and second distributions.

7 In yet another implementation, a system is provided. A region sampling  
8 module samples pixels in a first region within a tool impression in a digital image  
9 to determine a first distribution of a pixel property of the pixels in the first region.  
10 The region sampling module also samples pixels in a second region within the tool  
11 impression to determine a second distribution of the pixel property of the pixels in  
12 the second region. An editing module edits at least one pixel within the tool  
13 impression based on the first and second distributions.

14 Other implementations are also described and recited herein.

15 FIG. 1 illustrates a screenshot of an exemplary adaptive region editing tool.

16 FIG. 2 illustrates another screenshot of an exemplary adaptive region  
17 editing tool.

18 FIG. 3 illustrates yet another screenshot of an exemplary adaptive region  
19 editing tool.

20 FIG. 4 illustrates exemplary relationships among a tool impression on a  
21 digital image, region property distributions, and two edit profiles.

22 FIG. 5 illustrates exemplary operations for adaptive region editing.

23 FIG. 6 illustrates an exemplary architecture of an adaptive region editing  
24 tool.

25 FIG. 7 illustrates an exemplary edit profile applied to two pixel properties.

1        FIG. 8 illustrates exemplary classifications for distributions of multiple  
2 properties.

3        FIG. 9 illustrates an exemplary system useful for implementing an  
4 implementation of the present invention.

5        FIG. 10 illustrates exemplary pixel property distributions and an edit profile  
6 parameters relative to an example image and tool impression.

7        FIG. 11 illustrates other exemplary pixel property distributions and an edit  
8 profile parameters relative to an example image and tool impression.

9        FIG. 12 illustrates other exemplary pixel property distributions and an edit  
10 profile parameters relative to an example image and tool impression.

11       FIG. 13 illustrates other exemplary pixel property distributions and an edit  
12 profile parameters relative to an example image and tool impression.

13       FIG. 14 illustrates other exemplary pixel property distributions and an edit  
14 profile parameters relative to an example image and tool impression.

15       FIG. 15 illustrates other exemplary pixel property distributions and an edit  
16 profile parameters relative to an example image and tool impression.

17       FIG. 16 illustrates other exemplary pixel property distributions and an edit  
18 profile parameters relative to an example image and tool impression.

19       FIG. 17 illustrates other exemplary pixel property distributions and an edit  
20 profile parameters relative to an example image and tool impression.

21       FIG. 18 illustrates an example of hue modification.

22       An exemplary adaptive region editing tool samples one or more properties  
23 (e.g., colors, texture) of pixels in a digital image within different subdivisions of  
24 an editing tool impression to produce different property distributions. Generally,  
25 an editing tool combines an image region designation, typically via a user

1 interface, and an editing operation. An editing brush or wand represents an  
2 exemplary editing tool. The property distributions from each region are analyzed  
3 to identify different edit classes within the property space. The different edit  
4 classes are then used to apply an edit effect to the digital image within the tool  
5 impression (e.g., modifying hue property values of individual pixels). The edit  
6 classes may be represented by an edit profile in two or more dimensions (e.g.,  
7 applying to one or more pixel properties).

8 Editing tool impressions may be grouped and then adaptively edited in  
9 aggregate. For example, the tool impression may be stroked across the image or  
10 multiple tool impressions may be selected before the adaptive editing operation is  
11 executed. Adaptive editing involves altering an editable property of a pixel and  
12 may include without limitation background erasure, red-eye correction,  
13 retouching, masking, surface normal processing, modifying image effects,  
14 blurring, sharpening, adjusting contrast, and labeling of image areas and features.

15 FIG. 1 illustrates a screenshot 100 of an exemplary adaptive region editing  
16 tool. An image window 102 contains a display of a digital image 104. Selection  
17 of a background erasure tool button 106 initiates the exemplary adaptive region  
18 editing tool, designated by tool symbol 108. Operation of the exemplary tool is  
19 characterized by display of a tool impression 110 around the tool symbol 108 in  
20 the illustrated implementation. In one implementation, the tool impression 110  
21 defines the portions of the digital image to be analyzed during an adaptive region  
22 editing operation. The tool impression 110 may also be manipulated by the user  
23 within the digital image display, such as by use of a keyboard, mouse, stylus, or  
24 other input device. Manipulation may include without limitation changes in  
25 location, size, and orientation.

1 In FIG. 1, the tool impression 110 is illustrated as a circular region,  
2 although other shapes may be employed, including without limitation square  
3 regions, rectangular regions, and adjustable regions. The illustrated tool  
4 impression 110 defines two component regions: a central region 112, and an outer  
5 region between the central region 112 and the outer boundary of the tool  
6 impression 110. Furthermore, in various implementations, the dotted line  
7 boundary of the central region 112 may or may not be visible to the user. In  
8 alternative implementations, tool impression regions and component regions of  
9 other shapes and orientation may be employed, including rectangular regions,  
10 triangular regions, regions of different shapes, adjacent regions (as opposed to  
11 concentric), overlapping regions, and non-adjacent regions. In addition, more than  
12 two regions may be employed.

13 In alternative implementations, the size, shape, location, and orientation of  
14 the component regions or of the tool impression itself may be dependent on a user-  
15 setting or on image content found under the tool impression. For example, given a  
16 tool impression size setting (e.g., a BrushSize), a size of a central component  
17 region (e.g., having a SampleSize) may be set according to the following  
18 configuration (units in pixels):

19 If BrushSize < MinBrushThreshold then,

20 SampleSize = MinSampleSize

21 Else if  $\text{MinBrushThreshold} \leq \text{BrushSize} \leq \text{MaxBrushThreshold}$  then

22 SampleSize = BrushSize \* ScalingFactor

23 Else if BrushSize > MaxBrushThreshold then

24 SampleSize = MaxSampleSize

25 Alternatively, the BrushSize and/or SampleSize may be dynamically sized  
based on the values of pixel properties found in the tool impression. For example,  
the SampleSize may be automatically set by incrementally increasing the Sample



1 Size until the width of a color difference distribution in the central region  
2 increases beyond a given threshold width, which would suggest that the central  
3 region has grown to include a portion of the object's color. Other approaches for  
4 dynamically adapting the size, shape, and orientation of the component regions  
5 based on image content are also contemplated.

6 The multiple tool impression regions allow a user to provide intelligent guidance  
7 for the selection of how and whether an individual pixel is edited. For example, as  
8 discussed, a user may wish to erase the background of an image, leaving the  
9 human subject (i.e., the foreground object) unchanged. In the illustration of  
10 FIG. 1, the tool impression 110 is placed in the digital image 104 so as to erase the  
11 pixels comprising the sky while leaving the pixels comprising balloon and basket  
12 relatively unchanged. Accordingly, the user can communicate guidance for this  
13 selection to the tool by including in the central region 112 only colors to be erased  
14 and including in the outer region some portion of object colors. Given this region-  
15 based information, the tool can then discern through classification which pixels to  
16 edit and how to edit them.

17 Examples of editing may include without limitation modifying image-  
18 related pixel properties, such as color, transparency, etc., but may also include  
19 modifying existing non-image-related pixel properties or adding additional  
20 properties (e.g., labels or properties applied to certain areas of the digital image).  
21 Various pixel properties and color spaces may also be employed, singly or in  
22 combination, in other implementations, including CMYK, Lab, and other color  
23 space values, transparency, brightness, hue, saturation. Furthermore, non-image  
24 pixel properties are also suitable, including a membership associated with a pixel,  
25

1 a feature vector (e.g., a texture descriptor or a color histogram of an image region),  
2 a height map, a depth map in a scene, a displacement vector, etc.

3 It should also be understood that different color spaces may be used for the  
4 individual adaptive editing sub-operations, including determining component  
5 region characteristics, determining an edit profile, performing the edit, and  
6 generating display characteristics. For example, the edit profile may be based on  
7 color distributions in the regions while the editing operation may perform a  
8 “selective restoration” action in which a previously executed editing operation is  
9 selectively “undone” (e.g., by performing a generally opposite editing action)  
10 according to the edit profile. In this manner, aspects of the image region may be  
11 selectively restored to a previous state (i.e., by selectively undoing a previously  
12 executed fill operation).

13 In the illustrated implementation, the appearance of a pixel is characterized  
14 by at least four values: three color channel values (red, blue, and green or RGB)  
15 and an opacity (or conversely, transparency) channel value, which defines the  
16 opacity of the pixel. The color channel values may be combined in a variety of  
17 ways to represent a single color selected from a given color space. The opacity  
18 channel value represents a degree by which an object obscures another object  
19 beneath it. For example, an object with 100% opacity completely conceals an  
20 object beneath it. In contrast, an object with 0% opacity is completely transparent  
21 and allows an object beneath it to be totally unobscured. An object with an  
22 opacity value between 0% and 100% is partially transparent. Specifically, the  
23 opacity of an object is defined by the opacity of the individual pixels that comprise  
24 it. (In FIG. 1, the checkerboard pattern visible in the image represents a region of  
25 pixels exhibiting some level of transparency.)

1        Accordingly, the exemplary adaptive editing tool of FIG. 1 samples pixel  
2 properties within each of the component regions of its tool impression to  
3 determine pixel property distributions within each component region. For  
4 example, in the illustrated tool example, the colors of each pixel (or of some  
5 subset of individual pixels) within the central region 112 are identified. In  
6 addition, the colors of pixels within the outer region are sampled. From these  
7 pixel color samples, the tool computes pixel color distributions associated with  
8 each region. It should be understood that property distributions may be quantized  
9 to any chosen degree and may represent, without limitation, frequencies of pixels  
10 having given property values, an integration of such frequencies, or some other  
11 mathematical derivation of the property frequencies. In addition, in some  
12 implementations, property distributions may be represented by histograms or  
13 hierarchies.

14        Sampling a pixel generally refers to the process of reading a pixel property  
15 for the purpose of analysis or classification. Furthermore, a pixel property may  
16 include an overall image, object or region property that is associated with a given  
17 pixel (e.g., an object label). Sampling may also be performed on all pixels in a  
18 given region or on a representative fraction of the pixels in the region, which may  
19 be considered sparse sampling.

20        Given the pixel property distributions of the multiple component regions in  
21 the tool impression, the tool determines edit classes in the pixel property space  
22 (e.g., the color space) to which an editing effect may be assigned. For example,  
23 edit classes may be specified as: (1) an Erasure Class - a class of colors to be  
24 completely erased; (2) a Partial Erasure Class - a class of colors to be partially  
25 erased; and (3) an Unchanged Class - a class of colors to be left unchanged.

1 However, in alternative implementations, it should be understood that any number  
2 of classes and editing effects may be controlled by the edit profile.

3 In the illustrated example, each pixel within the tool impression 110 is  
4 classified into one or more defined edit classes based on the pixel property value  
5 associated with the pixel. In some implementations, the defined edit classes may  
6 model classified pixel property distributions within a tool impression or may be  
7 derived from such classified pixel property distributions (e.g., computed based on  
8 a mathematical transform of the distributions). In one implementation, the editing  
9 effects are assigned to individual edit classes by way of an edit profile, as  
10 discussed with regard to FIG. 4. In alternative implementations, multiple editing  
11 effects may also be combined in more than one edit class (e.g., as a result of  
12 overlapping edit classes), or edit classes and membership therein may be  
13 determined by a probability function, so as to provide somewhat fuzzy or non-  
14 discrete edit class boundaries in the property space.

15 Accordingly, an edit profile may be used to apply one edit effect to one edit  
16 class of pixels within the image and a different edit effect to another edit class of  
17 pixels. In contrast, multiple edit effects for multiple edit classes are also  
18 contemplated. An edit effect may modify one pixel property of one edit class and  
19 the same or different pixel property of a second edit class. The pixel property  
20 being modified may be the same as or different from that used to segregate pixels  
21 into edit classes. The transition between different edit effects may be abrupt or  
22 gradual, with some pixels receiving an intermediate effect. Pixels receiving an  
23 intermediate effect may be of a separate edit class or may represent regions of  
24 overlap of class probability or class membership. The edit profile may determine  
25 an effect to be applied in absolute terms, for instance by prescribing a new value

1 for a pixel property to replace the old, such that the new value (i.e., a replacement  
2 value) depends solely on the profile. The edit profile may also designate values of  
3 a new pixel property not previously associated with pixels within the image.  
4 Alternatively, the edit profile may represent a mathematical transformation  
5 between old and new property values of a pixel, such that the resulting image  
6 property depends both on the edit profile and on the original pixel property value.  
7 For instance, the edit profile may represent a scaling factor for an existing image  
8 property. More complex transformations are also contemplated wherein the new  
9 pixel property is expressed as a parameterized function of the original property  
10 with parameter values determined by the value of the edit profile.

11 In the illustrated example, erasure may be accomplished by setting the  
12 opacity channel values of pixels within the tool impression 110 having color  
13 values within the erasure class to 0% opacity, setting the opacity channel values of  
14 pixels within the tool impression 110 having color values within the partial erasure  
15 class to some value between 0% and 100%, and setting the opacity channel values  
16 of pixels within the tool impression 110 having color values within the unchanged  
17 class to 100% opacity. (In one implementation, the opacity channel value of each  
18 pixel is scaled, not merely set, between 0% and 100% original opacity of the pixel  
19 in accordance with the edit profile, in order to accommodate pixels already having  
20 a non-zero opacity.) Alternatively, transparency may be scaled by a scaling factor  
21 derived from the edit profile. Furthermore, the edit profile may set the allowable  
22 extremes of the editing effect to factors different than 0% and 100%. As  
23 previously noted, erasure is only one example of the type of editing that may be  
24 selectively performed using implementations described herein.  
25

1        FIG. 2 illustrates another screenshot 200 of an exemplary adaptive region  
2 editing tool. Two tool impressions have been selected in sequence. The first tool  
3 impression (shown at image area 202) resulted in selective erasure of the area of  
4 sky while leaving the balloons and baskets at 204 relatively unchanged. A second  
5 tool impression is shown at 206 prior to the associated editing operation. In one  
6 implementation, a new edit profile is defined for each instance of a tool impression  
7 so as to provide one type of adaptive feature to the editing operation – the edit  
8 profile adapts to the image content under each tool impression.

9        FIG. 3 illustrates yet another screenshot 300 of an exemplary adaptive  
10 region editing tool. Multiple tool impressions have been selected and edited in  
11 sequence to leave the erased (i.e., checker-boarded) area 302. The illustrated  
12 digital image has a single layer. Thus, no image data is displayed in the erased  
13 area 302.

14        Individual controls and parameters of the adaptive tool illustrated in FIG. 3  
15 are described below:

16        ***Presets:*** specific combinations of the other brush settings that may be saved  
17 and retrieved for convenience

18        ***Brush tips:*** grayscale bitmaps acting as masks for the brush and  
19 determining which portion of the tip will have an effect and to what degree. Where  
20 the bitmap is black, the brush exerts a full effect, where it is white no effect, and  
21 where it is grey an intermediate effect.

22        ***Brush shapes:*** brush tip shape definitions based on a computed shape  
23 rather than a bitmap mask. The basic shapes are round or square and can be further  
24 modified by other controls.  
25

1       **Size:** the size of brush in pixels. For irregularly-shaped brushes, the size is  
2 the bounding box of the shape. For round brushes, it is the diameter and for square  
3 brushes the length of a side.

4       **Hardness:** the degree to which the effect of the brush is scaled outwards  
5 from the center of the brush. At maximum Hardness, the brush has the same  
6 magnitude of effect everywhere under the brush (subject to masking due to  
7 loading a Brush Tip). As Hardness is decreased, the maximal effect of the brush is  
8 more and more confined to the center of the brush and decreases progressively  
9 outwards.

10       **Step:** intervals for designating individual tool impressions in a brush path or  
11 stroke. As the brush is moved along a path or stroke, it puts down tool  
12 impressions at intervals determined by the Step. A Step of 100 corresponds to  
13 moving the brush a distance equal to 100% of its Size. Smaller values put down  
14 more frequent impressions and larger values less frequent impressions.

15       **Density:** the fraction of randomly selected pixels under the brush to which  
16 the brush effect will be applied. At the maximum Density of 100 all pixels under  
17 the brush are affected (subject to masking due to loading a Brush Tip); otherwise  
18 not all of the pixels are affected, with fewer being affected the lower the density  
19 setting.

20       **Thickness:** a deformation of a round or square brush to, respectively, an  
21 ellipse or rectangle. The shape is symmetrical at a Thickness of 100 and  
22 increasingly unsymmetrical as the Thickness is reduced.

23       **Rotation:** the degree to which the brush tip is rotated relative to the  
24 horizontal or vertical axis of the image.

1       **Opacity:** a measure of uniformly scaling applied to the effect of the brush.  
2 After the geometric scaling due to a Brush Tip mask and the Hardness and Density  
3 settings are completed, the overall resultant effect of the brush is scaled by the  
4 Opacity. At an Opacity of 100 the brush produces the full result, while at lower  
5 values it has progressively less effect.

6       The following additional controls are specific to an implementation of an  
7 adaptive background erasure tool:

8       **Tolerance:** a measure of how color differences in the image are converted  
9 into differing levels of transparency. This control can either be set manually or  
10 simply shows the automatically computed Tolerance value when the Auto  
11 (adaptive) Tolerance checkbox is enabled.

12       **Sharpness:** a measure defining how object edges are treated. At high  
13 Sharpness values the brush will produce sharp edges on objects, corresponding to  
14 an abrupt transition in Alpha from background to object. High settings can be  
15 useful for differentiating an object from a background of almost identical color.  
16 Low Sharpness values will produce a wide semi-transparent edge with a distinct  
17 gradient in Alpha. Low settings can be useful for defining objects blurred by  
18 motion.

19       **Sampling:** a designation of how the image is sampled to determine the  
20 background color, with the following choices. *Continuous* means that a new  
21 determination of background color is made at every brush impression. *Once* means  
22 that the background color is sampled only at the initial impression that starts a  
23 brush stroke. This can be used when the background is uniform and allows the  
24 center of the brush to subsequently pass over the object without ill effect.



1       **Limits:** a setting that places some constraints on the assignment of Alpha,  
2 which are described more fully in the next section. Briefly, the *Discontiguous*  
3 mode has no constraints, *Contiguous* requires Alpha to be non-decreasing  
4 outwards from the center of the brush. *Find Edges* uses edge information in Alpha  
5 assignment.

6       **Sample Merged:** a control of brush behavior for an image comprising  
7 multiple layers. When unchecked only the image data on the current layer are  
8 sampled. Otherwise, image data from all visible layers are used. This control may  
9 have no meaning when the image is not being sampled to determine the  
10 background color.

11       **Ignore Lightness:** When checked the lightness component of color is not  
12 considered when estimating color similarity. Instead only chrominance  
13 information is used. This can be useful when object and background differ  
14 significantly in the saturation or vividness of color.

15       **Limits:** The illustrated brush also has a Limits setting with these choices:  
16 Discontiguous, Contiguous and Find Edges, which place additional constraints on  
17 how Alpha is assigned. Note that here the constraints primarily affect the  
18 assignment of Alpha outwards from the center of the brush (a “fill” operation)  
19 additionally to the assignment of Alpha based on image color (using a tolerance).

#### 20       *Discontiguous*

21               In the Discontiguous mode, the Alpha values of all the pixels  
22 under the brush tip outline are assigned.

#### 23       *Contiguous*

24               In the case of the Contiguous setting the assignment of Alpha  
25 values additionally depends on distance from the center of the brush

1 as follows. Alpha is chosen to be a non-decreasing function  
2 outwards from the center of the brush towards the periphery of the  
3 brush.

4 FIG. 4 illustrates exemplary relationships among a tool impression 400 on a  
5 digital image 402, region property distributions 404, and two edit profiles 406  
6 and 408. The digital image 402 includes an object 410 (representing a general  
7 class of pixels that are not to be edited) and a background 412 (representing a  
8 general class of pixels that are to be edited). Both the object 410 and  
9 background 412 may include multiple colors, textures, opacity values, etc. The  
10 adaptive region editing tool evaluates the pixel properties within the tool  
11 impression 400 to determine how to edit the pixels therein. In one  
12 implementation, the user places the tool impression 400 such that the outer region  
13 includes pixels of the object 410 (see region portion 414) and the central  
14 region 416 includes pixels of the background 412 but substantially none of the  
15 pixels of the object 410 to guide the selective editing operation of the tool. In  
16 alternative implementations, the tool impression 400 need not include any portion  
17 of the object. For example, an adaptive retouch tool can selectively modify pixels  
18 corresponding to an edit class in the outer sampling region or selectively modify  
19 pixels corresponding to the edit class in the central sampling region, in each case  
20 the pixels in the remaining edit class(es) being substantially unchanged.

21 The properties of pixels within the central region 416 are sampled to  
22 produce a pixel property distribution 418, and the properties of pixels within the  
23 outer region are sampled to produce a pixel property distribution 420. In FIG. 4,  
24 the vertical axis of distributions 404 represents the number or fraction of pixels  
25 having a pixel property value. The sampled property values are likely to be

1 distributed throughout each region. For example, if the pixel property is color (or  
2 a color difference), it is reasonable to expect that the various pixels might exhibit  
3 some distribution of color throughout each region.

4 It should be understood, however, that although this description of FIG. 4  
5 relates to a single type of property (e.g., color) for each pixel, multiple properties  
6 may also be sampled to produce distributions of multiple properties for each pixel,  
7 such as values of multiple color channels, opacity, texture, etc. A given property,  
8 such as color, may have one or more dimensions. For example, color represented  
9 by an index into a color palette could be considered as one-dimensional because a  
10 single number – the palette index – is sufficient to define it. Alternatively, color  
11 defined by an RGB triplet of a pixel is three-dimensional, one dimension for each  
12 of red, green and blue. However, a pixel property may also be composite. For  
13 example, a given pixel may be associated with several numbers representing  
14 texture in some way and some values representing color. Therefore, a pixel  
15 property may represent at least: (1) a one-dimensional single property, (2) a multi-  
16 dimensional single property, or (3) a composite property.

17 In one implementation, a distribution of pixel properties of pixels in the  
18 central region is analyzed. An exemplary pixel property that may be employed in  
19 this analysis involves the color difference between the color of the pixel at the  
20 center of the central region 416 and the color of each other pixel in the central  
21 region 416. The background color  $C_B$  (comprising of the color components  $R_B$ ,  
22  $G_B$ , and  $B_B$ ) is taken to be the color of the center pixel of the central region 416.  
23 The color  $C_B$  may be used to establish the origin for the color difference axis and  
24 may be used for unblending pixels containing colors of both the object and the  
25 background. It should be understood that the color  $C_B$  may also be computed

1 using other algorithms, such as an average or median or mode color of pixels in  
2 the central region. (A vector median or mode returns some color that actually  
3 exists in the image.)

4 Differences in pixel properties, including multidimensional properties, may  
5 be computed in a variety of ways dictated by the nature of the intended editing  
6 operation. Distance  $Dist$  in  $p$  property space of dimension  $i$  may be expressed as a  
7 Minkowski sum or  $b$  norm:

$$Dist = \left[ \sum_i (p_i - p_{i0})^b \right]^{1/b}$$

10 where  $p_{i0}$  is a reference property value relative to which differences are computed.  
11 This reference property value may, for example, represent a background color.  
12 The exponent  $b$  may have a variety of integer or non-integer values (e.g., 1, 2, or  
13 4). When  $b$  is 1, for a vector valued property  $p_i$ , such as color represented by R, G  
14 and B,  $Dist$  is equivalent to a simple sum:

$$Dist = (R - R_0) + (G - G_0) + (B - B_0)$$

17 For  $b=2$ ,  $Dist$  is computed as the Euclidean distance:

$$Dist = \sqrt{(R - R_0)^2 + (G - G_0)^2 + (B - B_0)^2}$$

21 Distances may be computed with regard to sign, or using absolute values as  
22 in a Manhattan distance:

$$Dist = |R - R_0| + |G - G_0| + |B - B_0|$$

Distance calculations may also take into account property distributions (e.g., by means of a Mahalanobis distance). Furthermore, distances may be computed along individual property dimensions  $i$ , or subsets of these dimensions, and then combined into a single metric by mathematical transformation. For example, a minimum value may be selected:

$$Dist = MIN(|R - R_0|, |G - G_0|, |B - B_0|)$$

or a weighted combination value may be derived:

$$Dist = w_R(R - R_0) + w_G(G - G_0) + w_B(B - B_0)$$

wherein  $w_i$  represents the weight for the contribution of a property or property dimension  $i$ . Additionally, one subset of property dimensions may be treated differently from another as in:

$$Dist = MAX(|R - R_0|, |G - G_0|) + |B - B_0|$$

Distances along individual dimensions may be combined into a single metric by rule, including the use of conditional rules as, for example, in:

$$Dist = MAX(|R - R_0|, |G - G_0|) + |B - B_0| \quad \text{if } Sign(R - R_0) = Sign(G - G_0) \neq Sign(B - B_0)$$

$$Dist = MAX(|G - G_0|, |B - B_0|) + |R - R_0| \quad \text{if } Sign(G - G_0) = Sign(B - B_0) \neq Sign(R - R_0)$$

$$Dist = MAX(|B - B_0|, |R - R_0|) + |G - G_0| \quad \text{if } Sign(B - B_0) = Sign(R - R_0) \neq Sign(G - G_0)$$

Depending on editing needs, it may be advantageous to form a metric from only a subset of pixel properties or only from a subset of pixel property dimensions. In one implementation pertaining to a prototypical vector  $p_i$  in a property space of dimension  $i = 4$ , a distance may be expressed for example as:

$$Dist = [(p_1 - p_{10})^b + (p_3 - p_{30})^b + (p_4 - p_{40})^b]^{1/b}$$

Properties may also be transformed from one representation to another before computing a distance. For example, a color vector may be transformed to an opponent color representation such as  $L^*a^*b^*$  or into an approximately opponent representation such as:

$$RGB = (R + G + B)/3$$

$$RG = (R - G + 256)/2$$

$$BY = (2B - R - G + 512)/4$$

After such a transformation either all or some of the properties or property components may be used to compute a property difference. For example, an approximately lightness-independent color difference may be estimated using:

$$Dist = |RG - RG_0| + |BY - BY_0|$$

or:

$$Dist = \sqrt{(RG - RG_0)^2 + (BY - BY_0)^2}$$

It should be understood that when a distance is estimated for a composite pixel property, such as combination of color and texture, the form of the metric for each component property may be chosen independently and the metrics may be combined with different weights or by independent rules.

When using multidimensional property spaces, especially spaces of high dimension, it may be beneficial to include dimensionality reduction in the estimation of a distance in property space. Such dimensionality reduction may, for example, be achieved by employing only the leading or leading few principal components of the space. Alternatively, discriminant axes may be determined by a method such as linear or Fisher discriminant analysis. It is also possible to use principal component analysis in combination with discriminant analysis to discard insignificant property component vectors and then select a most discriminating vector from the remaining components. In yet another implementation, distributions may be fitted using mixture models, which may then be used to classify properties of the image.

A result of the described sampling operation in the central region is a pixel property distribution, such as is represented by distribution 418. Likewise, the outer region is also sampled to produce an outer region pixel property distribution 420. Other methods of determining pixel property distributions (e.g., color difference distributions) may be employed.

For an exemplary classification used in an adaptive background eraser, a three-dimensional color space has been converted into one-dimensional distance measurements. However, it is also possible to achieve this dimensionality

1 reduction in other ways. First, dimensionality may be reduced by using only the  
2 leading principal components of the pixel colors under the tool impression.  
3 Secondly, one or more discriminant axes may be selected using, for example,  
4 linear discriminant analysis. The Principal Component Analysis step and the  
5 Linear Discriminant Analysis step can also be combined.

6 As seen in distributions 404, the distributions associated with the two  
7 regions (i.e., the central region 416 and the outer region of the tool  
8 impression 400) overlap considerably. In the illustrated example, this is to be  
9 expected in that the outer region includes a large area that contains background  
10 pixels. However, other distribution combinations are also possible, including  
11 combinations in which the central and outer region do not overlap, combinations  
12 in which multiple modes exist (whether the distributions overlap or not), etc. A  
13 variety of distributions are shown in examples in later Figures.

14 Accordingly, a feature of the exemplary tool is to take advantage of the  
15 user's placement of the tool impression regions to discern an edit profile based on  
16 the property distributions within the two differently located regions. In one  
17 implementation, it is assumed that the property distribution of pixels in the central  
18 region 416 represents the properties that are to be edited. It is also assumed that  
19 the property distribution of pixels in the outer region includes some properties that  
20 are not to be edited. These two assumptions facilitate the classification of  
21 properties values into distinct edit classes. It should be understood, however, that  
22 the editing may be in the inverse sense.

23 In one implementation, the edit profile is determined through classification  
24 of the property distributions to identify properties in different edit classes. In an  
25 example of an adaptive background erasure tool, Alpha (or "transparency") values



1 may be derived according to a measure of similarity between the color of a pixel  
2 under the brush and an estimate of the background color under the brush. Pixels  
3 having colors that are very similar are fully erased (set to 100% transparency),  
4 whereas pixels having colors that are very different are not erased (no change to  
5 the transparency of the pixels). Intermediate color differences result in a partial  
6 erasure by changing the transparency of a pixel to some intermediate Alpha value  
7 – the greater the color difference, the greater the change in the intermediate Alpha  
8 value.

9 As shown in the edit profile 406, the pixel property range from the origin to  
10 parameter  $T_1$  represents a first edit class (in this example, generally representing  
11 the background), the range between parameter  $T_1$  and parameter  $T_2$  represents a  
12 second edit class (in this example, generally representing a probable mix of  
13 background and object), and the range greater than parameter  $T_2$  represents a third  
14 edit class (in this example, generally representing the object).

15 In computing the piece-wise linear example of edit profile 406, parameters  
16 characterizing the two pixel property distributions are computed. In one  
17 implementation, the mean color difference of the background distribution,  $Mean_B$   
18 and the variance of the background distribution,  $Var_B$ , are computed for  
19 determining the minimum tolerance parameter  $T_1$ . When  $Var_B/Mean_B$  exceeds a  
20 threshold  $Thresh_B$ ,  $T_1$  may be computed as:

$$T_1 = k \times \sqrt{Var_B}$$

23 where  $k$  is a constant inversely proportional to the Sharpness setting. If  
24 threshold  $Thresh_B$  is not exceeded, the sampling region may be iteratively  
25

1 expanded by a one-pixel wide contour about its periphery until this threshold is  
2 reached. Then  $T_1$  may be estimated as before or as:

$$T_1 = k \times Integ$$

3  
4  
5 where *Integ* is the color difference at a fixed fraction of the integral of the  
6 difference distribution, such as 90% for instance.

7 Once the minimum tolerance parameter  $T_1$  is computed based on the pixel  
8 property distribution in the central region 416 of the tool impression 400, the pixel  
9 properties under the outer region of the tool impression 400 are analyzed to  
10 determine the maximum tolerance parameter  $T_2$ . The pixels within this region are  
11 characterized by an unknown combination of pixel properties (e.g., of object and  
12 background colors). The pixel property distribution 420 for this region is filtered  
13 to exclude all pixels having pixel properties below the minimum tolerance  
14 parameter  $T_1$ . In one implementation, the mean color difference of the object  
15 distribution,  $Mean_O$ , and the variance of the object distribution,  $Var_O$ , are  
16 computed for determining the maximum tolerance parameter  $T_2$ . Up to a  
17 predefined magnitude of  $k$ ,  $T_2$  may be taken as:

$$T_2 = k \times MIN(Mean_O, \sqrt{Var_O})$$

18  
19  
20 and above this magnitude as:

$$T_2 = k \times \sqrt{Var_O}$$

21  
22  
23  
24 If  $Var_B/Mean_B$  is less than the threshold  $Thresh_B$ , the value of  $T_2$  may be  
25 adjusted as follows:

$$T_2 = T_1 + T_2$$

Additionally, when  $Var_O/Mean_O$  is less than a threshold,  $Thresh_O$ , but  $\sqrt{Var_O}$  exceeds a second threshold,  $ThreshV_O$ , the same adjustment may be performed. In general,  $T_1$  and  $T_2$  are constrained to represent non-null color difference ranges and to lie within their respective data ranges, and the value of  $T_1$  is further constrained to be less than that of  $T_2$ .

Other types of classification may be employed to define editing classes. For example, more conservative values of thresholds may be used, i.e. a minimum tolerance  $T_1$  and a maximum tolerance  $T_2$  to define edit classes of colors that have a very high probability of belonging to the background and the object respectively. These two conservative edit classes may be used as training sets for a classifier that categorizes the remaining colors.

Alternatively, any of several probability density functions may be fitted to the pixel property distribution of the central sampling region. By subtraction of the extrapolated distribution of the color differences in the outer region of the brush, a better estimate of true object colors may be obtained to act as a training set. Suitable exemplary trial distributions include without limitation the normal, binomial, Poisson, gamma and Weibull distributions. Additional exemplary distributions may be found in the Probability distribution section of Wikipedia ([http://en.wikipedia.org/wiki/Probability\\_distribution](http://en.wikipedia.org/wiki/Probability_distribution)). Such distributions may also be used for classification using Bayesian statistics. Because of the projection of color differences onto a single axis, several distinct colors may have color differences falling in a single histogram bin. The original colors contributing to each bin can be identified in the three-dimensional color space and the training set

1 member colors can be used for a classification of colors in three-dimensional  
2 space.

3 Exemplary methods for suitable for this classification are described in T.-S.  
4 Lim, W.-Y. Loh and Y.-S. Shih, "A Comparison of Prediction Accuracy,  
5 Complexity, and Training Time of Thirty-three Old and New Classification  
6 Algorithms", *Machine Learning Journal*, vol. 40, p. 203-229, 2000, and include  
7 categories such as decision tree approaches, rule-based classifiers, belief networks,  
8 neural networks, fuzzy & neuro-fuzzy systems, genetic algorithms, statistical  
9 classifiers, artificial intelligence systems and nearest neighbor methods. These  
10 techniques may employ methodologies such as principal component analysis,  
11 support vector machines, discriminant analysis, clustering, vector quantization,  
12 self-organizing networks and the like.

13 As an alternative to this form of classification, the color distributions of the  
14 central region and the outer region may be treated as two signals to be separated  
15 into background and object categories by the techniques of blind signal separation.  
16 Since the central region provides a background signal not significantly  
17 contaminated by object colors, this represents a more straightforward case of semi-  
18 blind signal recovery. An element of blind signal separation is the use of higher  
19 order statistical measures. Suitable methods for blind signal separation are  
20 described in J.-F. Cardoso, "Blind signal separation; statistical principles", *Proc.*  
21 *IEEE*, vol. 9(10), p. 2009-2025, 1998 and in A. Mansour, A. K. Barros and N.  
22 Ohnishi, "Blind Separation of Sources: Methods, Assumptions and Applications",  
23 *IEICE Trans. Fundamentals*, vol. E83-A, p. 1498-1512, 2000.

24 The edit profile relates an editing effect, designated  $\theta$ , to the different edit  
25 classes. For example, in an erasure operation, the editing effect may represent a

transparency value (i.e., the inverse of opacity). As such, edit profile 406 specifies a 100% transparency value for the edit class between 0 and parameter  $T_1$ , representing complete erasure of pixels having colors falling in this range. In contrast, the edit profile specifies an unchanged transparency value for the edit class greater than parameter  $T_2$ , representing no change to the pixels having colors falling in this range. For pixels having colors falling between parameter  $T_1$  and parameter  $T_2$ , edit profile 406 specifies a linear scaling of the transparency effect (e.g., transparency change decreases with increasing color differences).

It should also be understood that although the edit profiles of FIG. 4 are shown to be based on a single pixel property distribution, edit profiles may also be generated for multiple pixel properties and therefore are not limited to two dimensions. See FIG. 7 as an example of three dimensional edit profile applied to two pixel properties.

In alternative implementations, an edit profile need not be based on sharp thresholds, as shown by the sigmoidal edit profile 408, where a parameter can define a point of inflection or any other feature of an edit profile based on a pixel property. For example, the parameters  $T_1$  and  $T_2$  may be employed to define an edit profile that defines an Alpha value (i.e., defining transparency of a pixel) based on the tilde function:

$$\text{Alpha} = 1 - \exp \left\{ -T_1 \left[ \ln (\text{DIFF}_{\text{max}} / \text{DIFF}) \right]^{T_2} \right\}$$

where DIFF represents the color difference of a pixel under the tool impression,  $\text{DIFF}_{\text{max}}$  represents the maximum DIFF value of all pixels under the tool impression, and  $T_1$  and  $T_2$  are the parameters.

Sigmoidal functions may also be used, including without limitation the arctangent and the tangent functions, and an adaptation of the Weibull function, with the form:

$$\text{Alpha} = 1 - \exp\left[-(T_2 / \text{DIFF})^{T_1}\right].$$

Other exemplary sigmoidal functions that may be adapted to define edit profiles are listed below:

- $$\text{Alpha} = 1 - \left( \frac{2}{\left( 1 + e^{-\left( \frac{\text{DIFF}}{s} \right)} \right)} - 1 \right) = \frac{2e^{-\left( \frac{\text{DIFF}}{s} \right)}}{\left( 1 + e^{-\left( \frac{\text{DIFF}}{s} \right)} \right)}$$
- $$\text{Alpha} = 1 - \frac{s\text{DIFF}}{\sqrt{1+s^2\text{DIFF}^2}}$$

The parameter  $s$  defining these functions may be, for example, derived by centering the point of inflection between the parameters  $T_1$  and  $T_2$  and matching the slope of the line between the parameters  $T_1$  and  $T_2$ .

FIG. 5 illustrates exemplary operations 500 for adaptive region editing. A detection operation 502 detects selection of a tool impression relative to a digital image. The selection defines at least a first region (e.g., a central region within the tool impression) and a second region (e.g., an outer region within the tool impression) within the tool impression. It should also be understood that FIG. 5 generally describes sequential processing for an individual tool impression, but that the same sequence of operations may be applied after selection of multiple

1 tool impressions, such that an aggregate tool impression with an aggregate first  
2 region and an aggregate second region are processed.

3 Various configurations of tool impressions and subdivided regions thereof  
4 may be employed. In one implementation, the regions are differently-located  
5 subdivisions of the tool impression. For example, as shown in FIGs. 1-4, the  
6 regions may be concentric (e.g., central region and outer ring). Alternatively,  
7 other differently located subdivisions of the tool impression may be employed.  
8 For example, adjacent square regions may be useful for selectively editing square  
9 or rectangular features of a digital image. Likewise, the regions may even be non-  
10 adjacent or independent. For example, the first region may be placed in a static  
11 location in the background and the second region may be moved from location to  
12 location along the boundary of the object. Furthermore, more than two regions  
13 may also be employed as well as regions of differing shapes, sizes, and  
14 orientations.

15 A sampling operation 504 determines a property distribution for pixels  
16 within the first region. For example, for each pixel, a color difference value (e.g.,  
17 relative to a reference color value) may be used as the pixel property. Another  
18 sampling operation 506 determines a property distribution for pixels within the  
19 second region. A classification operation 508 defines an edit profile based on the  
20 pixel property distributions within the first and second region, which are located in  
21 somewhat different locations, at somewhat different orientations, or at somewhat  
22 different sizes and shapes. The classification operation 508 may also be aided by  
23 the assumption that one region contains some pixels that are not intended to be  
24 edited whereas the other region does not, or other contextual knowledge of the  
25 problem domain (e.g., red-eye correction).

1       The edit profile specifies the editing effects to be applied to pixels having  
2 properties in a given edit class. Accordingly, an editing operation 510 edits the  
3 pixels within the tool impression in accordance with the edit profile. For example,  
4 pixels in the erasure region may have their transparency values set to 100% while  
5 pixels in the partial erasure region may have their transparency values set to some  
6 value between 100% and 0% in accordance with the edit profile.

7       In one alternative implementation, different editing effects may be applied  
8 to pixels in different editing classes. For example, background pixels may be  
9 lightened while foreground or object pixels may be darkened, thereby adaptively  
10 adjusting contrast in the selected regions. In another example, existing opacity  
11 may be enhanced in an object region and reduced in a background region to  
12 provide adaptive adjustment of transparency contrast. An example of hue  
13 modification is shown in FIG. 18.

14       FIG. 6 illustrates an exemplary architecture of an adaptive region editing  
15 tool 600. In various implementations, modules may be structured as hardware or  
16 as software executing within a computer system. A digital image 602 may be  
17 represented by encoded data, such as in JPEG format, GIF format, or some other  
18 standard or proprietary format. The digital image 602 may be received from a  
19 persistent storage medium 604 (e.g., a magnetic or optical storage disk, or flash  
20 memory) or from a communications network 606 (e.g., the Internet or a local area  
21 network). The digital image 602 is transferred to another storage medium, such as  
22 memory 608 for access by the editing tool 600.

23       A user interface 610 can access the digital image 602 from the memory 608  
24 and display it to a user. For example, FIGs. 1-4 illustrate user interface  
25 screenshots of an exemplary adaptive region editing tool. The user interface 610



1 may also receive input from the user, relative to the digital image, instructing that  
2 assorted editing operations be performed on an area of the digital image 602 as  
3 well as parameters for such operations, such as the location and dimensions of the  
4 tool impression.

5 The input is received by an impression definition module 612, which  
6 accesses the memory resident representation of the digital image 602, and  
7 communicates the location and dimensions of the tool impressions as well as the  
8 two or more subdivisions (i.e., component regions) therein. The subdivisions are  
9 differently located within the tool impression and therefore have at least one pixel  
10 difference between them. In one implementation, this impression definition  
11 occurs for each tool impression, whereas in other implementations, this impression  
12 definition may be applied to multiple tool impressions (e.g., to multiple distinct  
13 tool impressions or to a stroke of the tool impression across the image, which is  
14 interpreted as multiple individual tool impressions). For such aggregate tool  
15 impressions, first and second component regions may be determined in a variety  
16 of ways. One example is that tool impression regions that overlap a sampling  
17 (e.g., background) component region of any tool impression in the set are  
18 considered sampling component regions while all other tool impression regions  
19 are considered foreground component regions.

20 A region sampler module 614 samples the properties of the pixels in each  
21 region to produce a property distribution for each region. An edit profile  
22 definition module 616 classifies the property distributions of the regions to  
23 determine a number of edit classes to which an editing effect is applied differently.  
24 Given the edit classes, an editing module 618 applies the edit classes to individual  
25 pixels within the tool impression. Therefore, using the region erasure example,

1 pixels having property values falling into the erasure class have their transparency  
2 values set to 100%, pixels having properties values falling into the partial erasure  
3 class have their transparency values set to a tapered value between 100% and 0%,  
4 and pixels having properties values falling into the unchanged class have their  
5 transparency values unchanged. The editing changes are saved to memory, from  
6 which the user interface 610 can retrieve the updated digital image data and  
7 display the changes to the user.

8 FIG. 7 illustrates an exemplary edit profile 700 applied to two pixel  
9 properties *a* and *b*. In the edit profile 700, the pixel property *a* is shown along  
10 axis 702, the pixel property *b* is shown along axis 704, and the transparency  
11 editing effect is shown along axis 706. An edit profile component 708 applies to  
12 the pixel property *a*, and another edit profile component 710 applies to the pixel  
13 property *b*.

14 Combining the two edit profile components 708 and 710 results in a  
15 process of classifying the properties in two dimensions, as illustrated by the top  
16 view 712. The oval region between the origin and oval 714 represents the edit  
17 class of total erasure, such that pixels classified within the corresponding edit class  
18 have their transparencies set to 100%. The region between the oval 714 and the  
19 oval 716 represents the region of partial erasure, such that pixels classified within  
20 the corresponding edit class have their transparencies set to 100%. The region  
21 outside the oval 716 represents the region of no erasure, such that pixels classified  
22 within the corresponding edit class have their transparencies unchanged. The  
23 oval 718 represents the 50% point in the tapering of editing effect as applied to  
24 each property. It should be understood that a variety of pixel properties, such as  
25 lightness, hue, color, etc. could be additionally or alternatively modified.

1        FIG. 8 illustrates exemplary classifications for distributions of multiple  
2 properties. The classification 800 shows data points for pixels based on their  
3 values of the properties *a* and *b*. For example, the data point 802 represents a  
4 pixel having positive values for *a* and *b*, where its location relative to the edit  
5 classes suggests that its transparency will be set slightly greater than 0%, whereas  
6 the pixel represented by data point 804 will have its transparency unchanged.

7        The classification 800 shows that most of the background pixels (illustrated  
8 with black dots) are classified as within the erasure class threshold 806 or the  
9 partial erasure class threshold 808; whereas most of the foreground (e.g., object)  
10 pixels are classified as outside the partial erasure threshold 808.

11        FIG. 9 depicts an exemplary general purpose computer capable of  
12 executing a program product. One operating environment in which the described  
13 system is potentially useful involves the general purpose computer, such as shown  
14 as computer 913. In such a system, data and program files may be input to the  
15 computer, including without limitation by means of a removable or non-removable  
16 storage medium or a data signal propagated on a carrier wave (e.g., data packets  
17 over a communication network). The computer 913 may be a conventional  
18 computer, a distributed computer, or any other type of computing device.

19        The computer 913 can read data and program files, and execute the  
20 programs and access the data stored in the files. Some of the elements of an  
21 exemplary general purpose computer are shown in Figure 9, wherein a  
22 processor 901 is shown having an input/output (I/O) section 902, at least one  
23 processing unit 903 (e.g., a microprocessor or microcontroller), and a memory  
24 section 904. The memory section 904 may also be referred to as simply the  
25

1 memory, and may include without limitation read only memory (ROM) and  
2 random access memory (RAM).

3 A basic input/output system (BIOS), containing the basic routines that help  
4 to transfer information between elements within the computer 913, such as during  
5 start-up, is stored in memory 904. The described computer program product may  
6 optionally be implemented in software modules, hardware modules, or firmware  
7 loaded in memory 904 and/or stored on a configured CD-ROM 908 or storage  
8 unit 909, thereby transforming the computer system in Figure 9 to a special  
9 purpose machine for implementing the described system.

10 The I/O section 902 is connected to keyboard 905, display unit 906, disk  
11 storage unit 909, and disk drive unit 907, typically by means of a system or  
12 peripheral bus (not shown). The system bus may be any of several types of bus  
13 structures including a memory bus or memory controller, a peripheral bus, and a  
14 local bus using any of a variety of bus architectures.

15 Generally, in contemporary systems, the disk drive unit 907 is a CD-ROM  
16 drive unit capable of reading the CD-ROM medium 908, which typically contains  
17 programs 910 and data. Computer program products containing mechanisms to  
18 effectuate the systems and methods in accordance with the present invention may  
19 reside in the memory section 904, on a disk storage unit 909, or on the CD-ROM  
20 medium 908 of such a system. Alternatively, disk drive unit 907 may be replaced  
21 or supplemented by a floppy drive unit, a tape drive unit, or other storage medium  
22 drive unit. The network adapter 911 is capable of connecting the computer system  
23 to a network via the network link 912.

24 The drives and their associated computer-readable media provide  
25 nonvolatile storage of computer-readable instructions, data structures, program

1 modules and other data for the computer 913. It should be appreciated by those  
2 skilled in the art that any type of computer-readable media which can store data  
3 that is accessible by a computer, such as magnetic cassettes, flash memory cards,  
4 digital video disks, random access memories (RAMs), read only memories  
5 (ROMs), and the like, may be used in the exemplary operating environment.

6 The computer 913 may operate in a networked environment using logical  
7 connections to one or more remote computers. These logical connections are  
8 achieved by a communication device 911 (e.g., such as a network adapter or  
9 modem) coupled to or incorporated as a part of the computer 913; the described  
10 system is not limited to a particular type of communications device. Exemplary  
11 logical connections include without limitation a local-area network (LAN) and a  
12 wide-area network (WAN). Such networking environments are commonplace in  
13 office networks, enterprise-wide computer networks, intranets and the Internet,  
14 which are all exemplary types of networks.

15 In an exemplary implementation, an impression definition module, a region  
16 sampler module, an edit profile definition generator, and an editing module may  
17 be incorporated as part of the operating system, application programs, or other  
18 program modules executed by the CPU 903. The digital image, the property  
19 distributions, thresholds, edit profiles, and edit classes may be stored as program  
20 data in memory section 904, on a disk storage unit 909, or on the CD-ROM  
21 medium 908.

22 FIG. 10 illustrates exemplary pixel property distributions 1000 and an edit  
23 profile parameters 1002 relative to an example image 1004 and tool  
24 impression 1006.  
25

1 FIG. 11 illustrates other exemplary pixel property distributions 1100 and an  
2 edit profile parameters 1102 relative to an example image 1104 and tool  
3 impression 1106.

4 FIG. 12 illustrates other exemplary pixel property distributions 1200 and an  
5 edit profile parameters 1202 relative to an example image 1204 and tool  
6 impression 1206.

7 FIG. 13 illustrates other exemplary pixel property distributions 1300 and an  
8 edit profile parameters 1302 relative to an example image 1304 and tool  
9 impression 1306.

10 FIG. 14 illustrates other exemplary pixel property distributions 1400 and an  
11 edit profile parameters 1402 relative to an example image 1404 and tool  
12 impression 1406.

13 FIG. 15 illustrates other exemplary pixel property distributions 1500 and an  
14 edit profile parameters 1502 relative to an example image 1504 and tool  
15 impression 1506.

16 FIG. 16 illustrates other exemplary pixel property distributions 1600 and an  
17 edit profile parameters 1602 relative to an example image 1604 and tool  
18 impression 1606.

19 FIG. 17 illustrates other exemplary pixel property distributions 1700 and an  
20 edit profile parameters 1702 relative to an example image 1704 and tool  
21 impression 1706.

22 FIG. 18 illustrates an example of hue modification. An image 1800  
23 includes three vertical stripes on a cyan field: red, green, and blue (from left to  
24 right). An image 1802 also has three vertical stripes on a cyan field: red, green,  
25 and blue (from left to right). An adaptive hue modification tool, the tool

1 impression of which is show as circle 1804, has been used to selectively change  
2 the hue in a portion of each of the vertical stripes in image 1802 as the tool  
3 impression 1804 overlaps two color regions. The resulting modified portion of  
4 each vertical stripe is generally a yellow hue, with unchanged saturation and  
5 preservation of the original varying levels of lightness.

6 The embodiments of the invention described herein are implemented as  
7 logical steps in one or more computer systems. The logical operations of the  
8 present invention are implemented (1) as a sequence of processor-implemented  
9 steps executing in one or more computer systems and (2) as interconnected  
10 machine modules within one or more computer systems. The implementation is a  
11 matter of choice, dependent on the performance requirements of the computer  
12 system implementing the invention. Accordingly, the logical operations making  
13 up the embodiments of the invention described herein are referred to variously as  
14 operations, steps, objects, or modules. In addition, data structures may be  
15 represented by individual objects, defined data structures, individual database  
16 tables, or combinations of associated database tables. A data field of a data  
17 structure may be represented or referenced by one or more associated database  
18 table fields.

19 The above specification, examples and data provide a complete description  
20 of the structure and use of exemplary embodiments of the invention. Since many  
21 embodiments of the invention can be made without departing from the spirit and  
22 scope of the invention, the invention resides in the claims hereinafter appended.  
23  
24  
25